Build a projBox Controller

Written By: Joseph Gray

TOOLS:

- Electrical Tape (1)
- Needle Nose Pliers (1)
- Super glue (1)
- Wood glue (1)
- projBoxTester Processing Sketch (1)
  Download at: http://projbox.org/software/
- small screw-driver (1)

PARTS:

- projBox (1)
  Pre-made here: http://projbox.org/get/ or cut-out your own: http://projbox.org/templates/
- Arduino microcontroller (1)
- tiny screws (2)
- Prototyping PC board (1)
- 10k potentiometers (4)
- SPST toggle switch (4)
- LED (4)
- USB cable (1)
  mini male to standard male
- various wires (1)

SUMMARY

The finished projBox controller provides physical inputs for your software in addition to the computer's conventional keyboard and mouse. The combination of open-source enclosure schematics and the use of standard electronic components provides an infinitely

customizable prototyping platform.

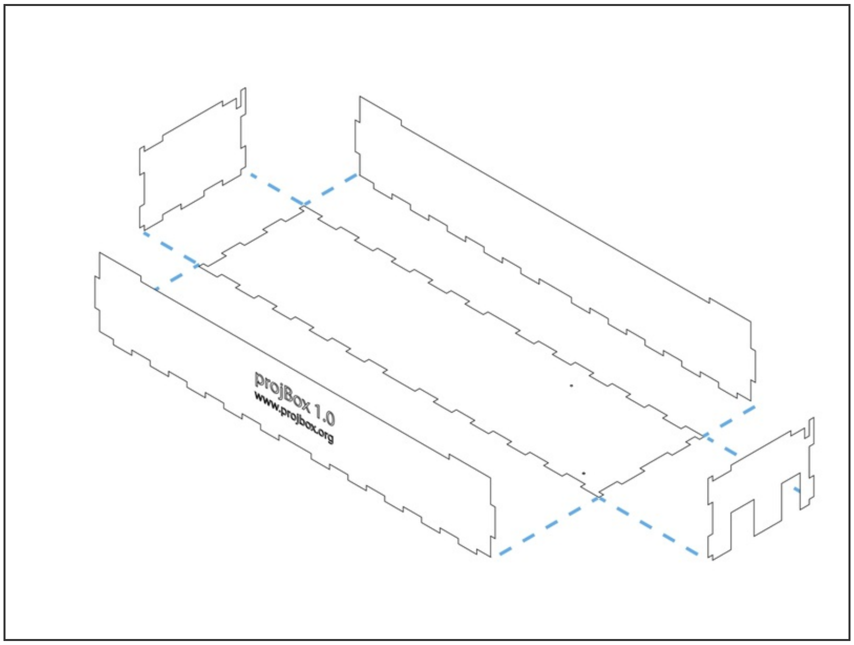
The projBox is specifically designed to integrate with software written in Processing, the IDE that the Arduino's coding environment is based on. The switches and knobs on the box can be mapped to any function; for instance, turning a knob might change the line color in a drawing tool.

Four rows of knobs, switches and LEDs can be attached to the lid and used to control custom software via the USB port on the Arduino. Other configurations, such as replacing the knobs with IR or proximity sensors, are also possible with some modification.

A projBox Kit is available from the Maker Shed for those entirely new to electronics. The kit is designed to supplement a video course published by O'Reilly Media called Processing and Arduino in Tandem. The series shows, in detail, how to assemble the projBox as an example Arduino project and how to write custom applications in Processing. The kit contains everything you need with all of the parts pre-soldered and ready to go.

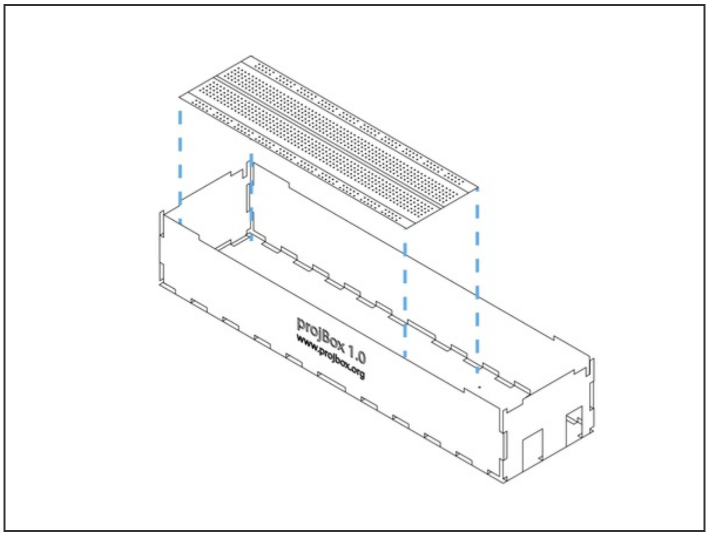
If you already have a grasp on both Arduino and Processing you can use your own parts or simply riff off of this technique as a way to combine commonly obtainable technologies for projects involving custom human/computer interaction. In other words: modify as you see fit.

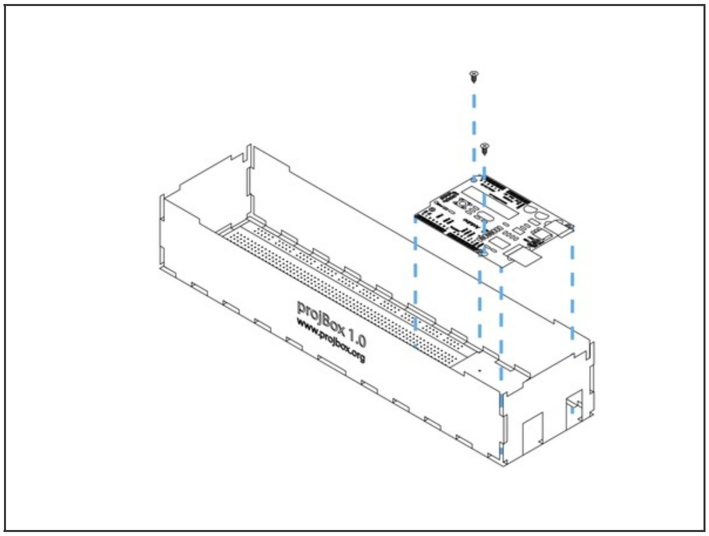
Step 1 — Build the box



- Try dry-assembling the box first to see how everything fits together. The smaller slots on the top of the side pieces should both be towards the back of the box and the ports should be on the right side. Also, one of the screw holes on the bottom should be to the lower right. Once everything fits, glue it up. Wood glue works really well with the plywood.

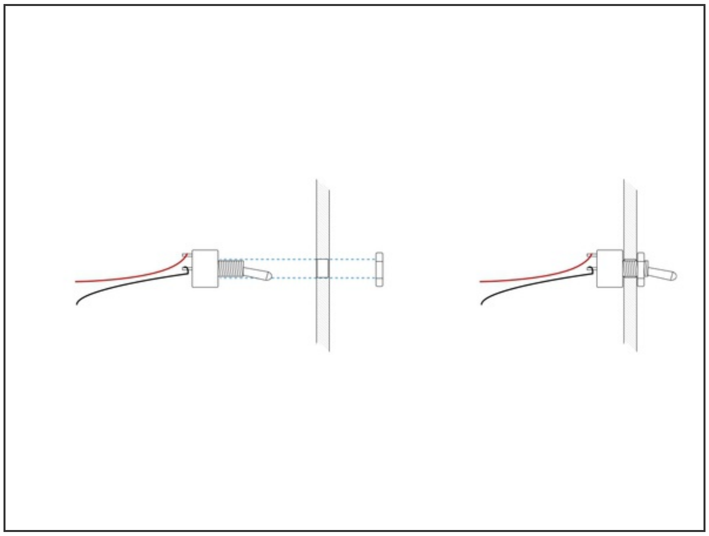
Step 2 — Add the protoboard and Arduino

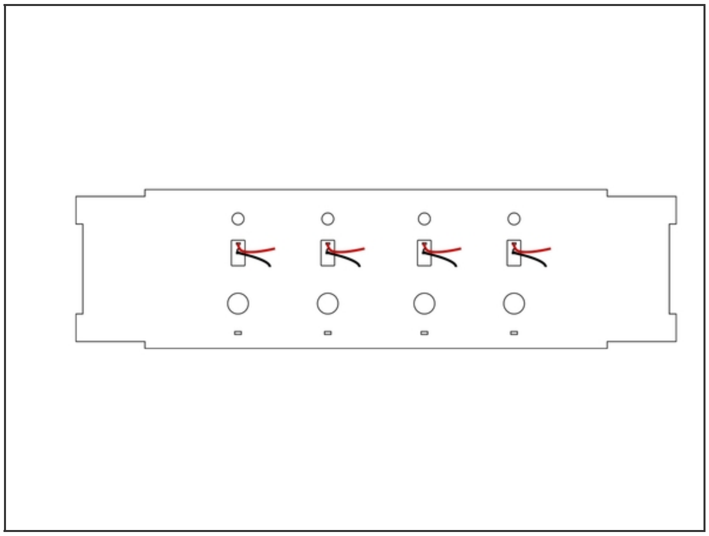




- Remove the sticker backing from a standard-size protoboard, then place this in the box snug against the end opposite the ports. Try to leave a small gap on each of the long sides of the protoboard. This gap can be useful for tidying up wires. Press the protoboard onto the base of the box. The adhesive on these is quite strong so be careful when lowering it into position. You pretty much have one shot at it before it sticks permanently.
- Next, tip the Arduino into place with the ports in first, then lowering down the side towards the protoboard. Two small screws will attach the Arduino through two small holes in the base of the box.

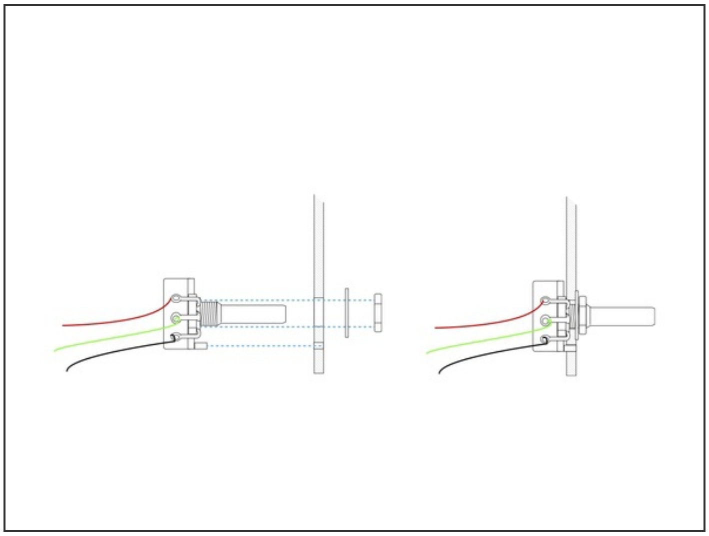
Step 3 — Attach the switches

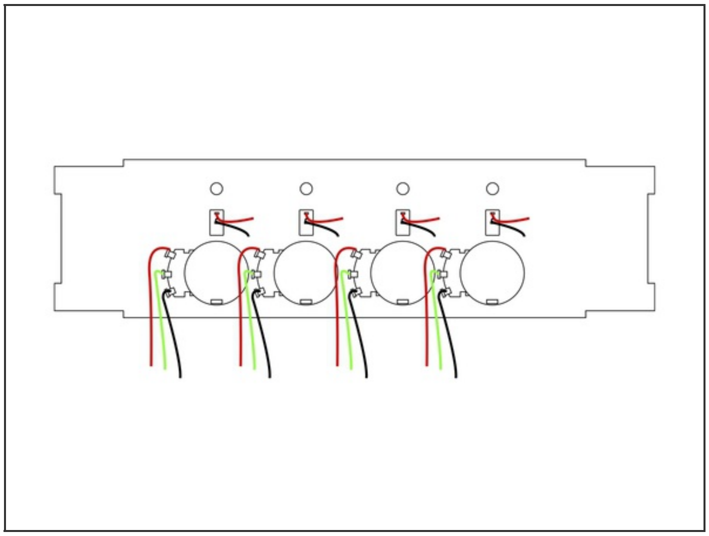




- Attach the switches to the lid using a nut and washer. Needle-nose pliers are useful; try holding the switches in place with the pliers by clamping down on the rectangular section, then turning both the lid and switch a bit at the same time to twist them in. After it's in snug, attach a hex nut on the opposite side. Do this for each of the four switches.
- After they're all in you may want to gently curl the lead wires out of the way to make attaching the other components easier.

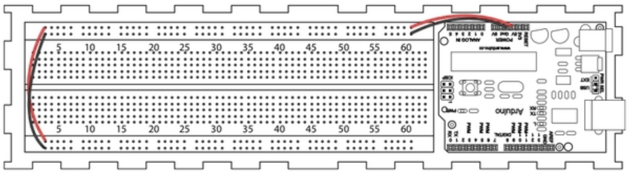
Step 4 — Attach the knobs





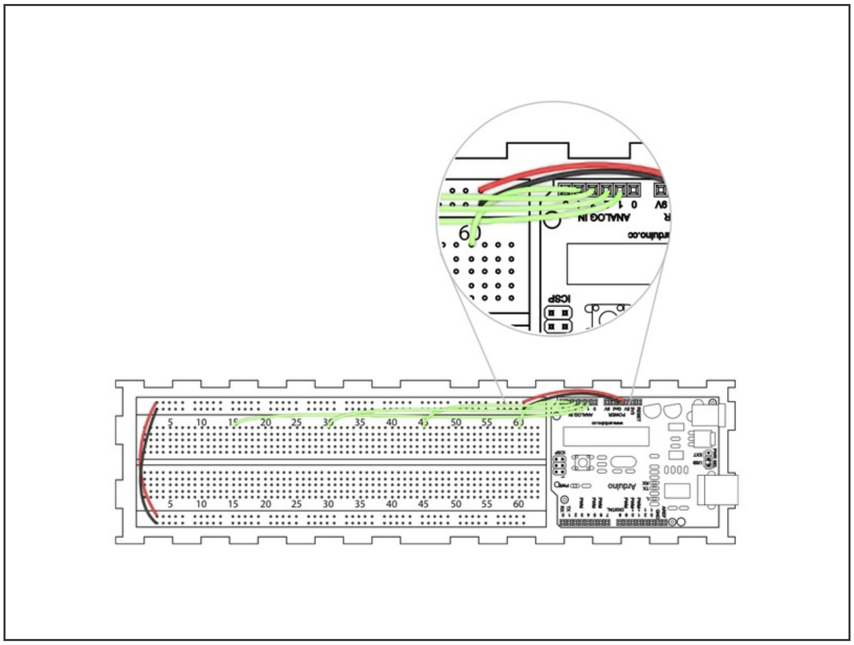
- Attach the knobs to the lid. Press them straight in with the set pin and main shaft both at a 90º angle to the lid. Then attach the washer and hex nut onto the shaft threading. The projBox lid has spaces for four knobs as well.

Step 5 — Hook up power



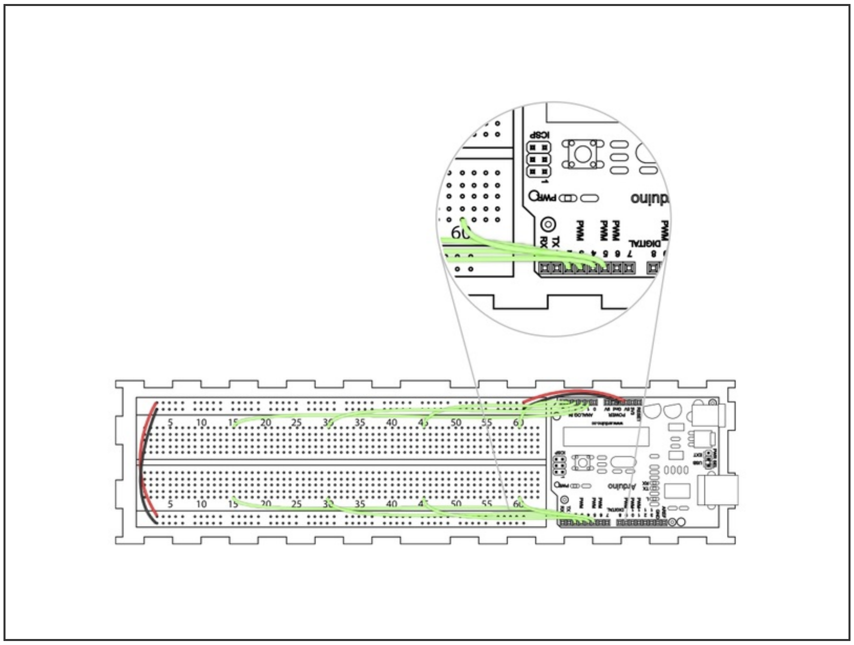
- Inside the box run a red wire from 5V on the Arduino to the top red rail on the protoboard. Run a black wire alongside it from one of the GND pins to the top blue rail.
- On the other end of the protoboard jump a red wire from the top red rail to the bottom one. Do this also with another black wire jumping from the top to bottom blue rails. This provides power for components connected nearer to that side of the box.
- Needle-nose pliers will be very useful throughout the process of hooking up the jumper wires inside the box, and then later connecting the components on the lid to the protoboard. Many times a wire can be coerced into place by holding the wire with the pliers just above the stripped part being inserted into a pin hole.

Step 6 — Connect jumper wires for the knobs



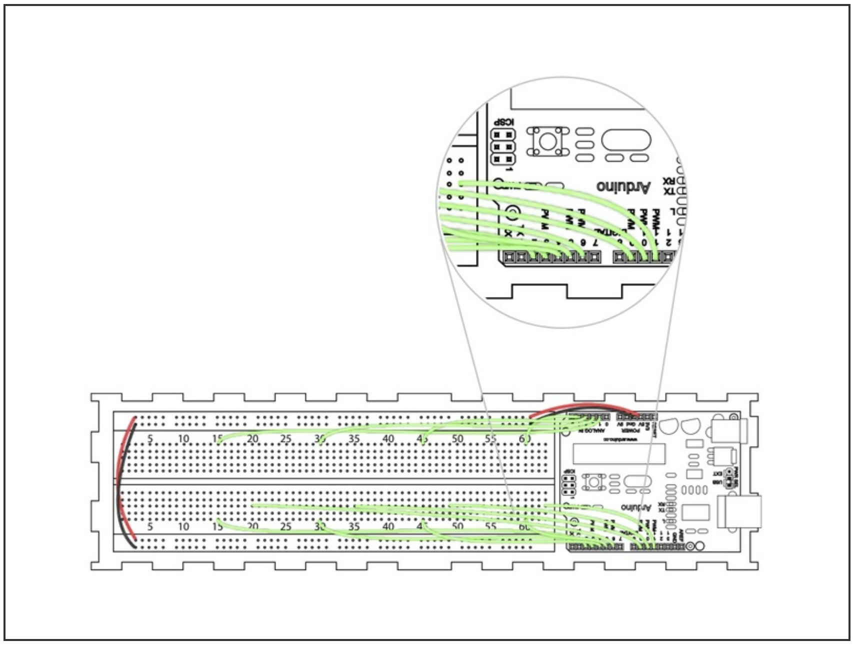
- Run green wires from the ANALOG IN pins 1 through 4 to the vertical rows on the top half of the protoboard, spacing every 15 rows, starting at row 15, to align the connections closer to the knobs on the lid. These wires will be used to connect the knobs' green (middle) output wires to the Arduino.

Step 7 — Connect jumper wires for the switches



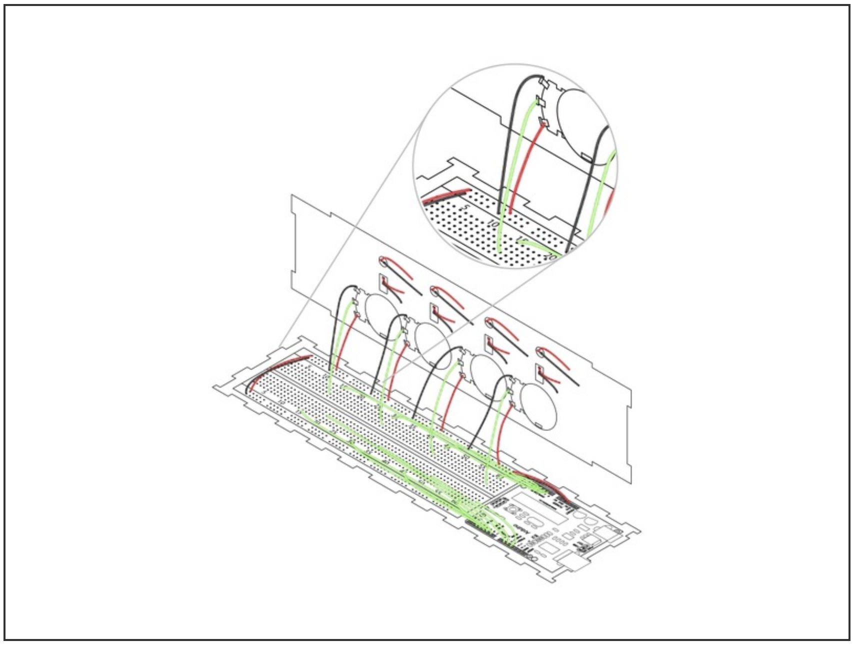
- Run green wires for the switch outputs from the bottom rows of the protoboard to the Arduino. Again, place them every 15 rows, starting from row 15, but this time connect the wires to Arduino's DIGITAL pins 2 through 5.
- If you allowed for a gap along the long sides of the protoboard try gently pushing the wires into the gap to tidy them up on both the top and bottom.

Step 8 — Connect jumpers for the LEDs



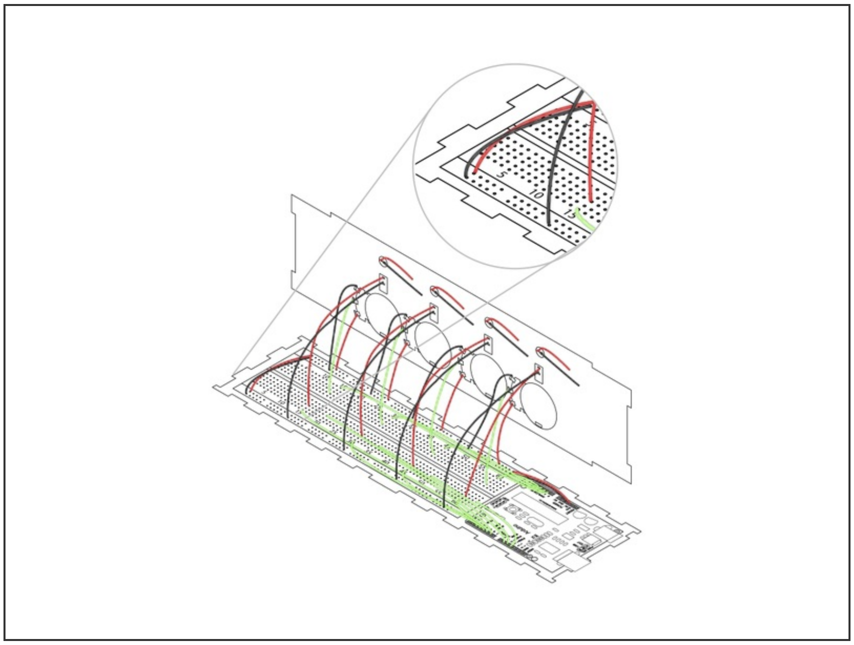
- The LEDs, in order to have variable output, must be connected to the DIGITAL pins marked PWM. PWM stands for "Pulse Width Modulation" and is a way to create the illusion of variable voltage being output to simple analog components by switching between high and low output levels at various speed intervals. More about PWM can be found at arduino.cc along with all the other specifics about the Arduino pins and their behavior.
- Run the green jumper wires for the LEDs every 15 rows starting at row 20 to stagger them against the wires for the switches. Also, connecting them a few spaces above the switch connections helps differentiate them. Connect wires from row 20 to DIGITAL 6, row 35 to DIGITAL 9, row 50 to DIGITAL 10 and row 63 to DIGITAL 11.
- With all the wires connected between the protoboard and the Arduino you're ready to connect the components on the lid.

Step 9 — Hook up the knobs to the protoboard



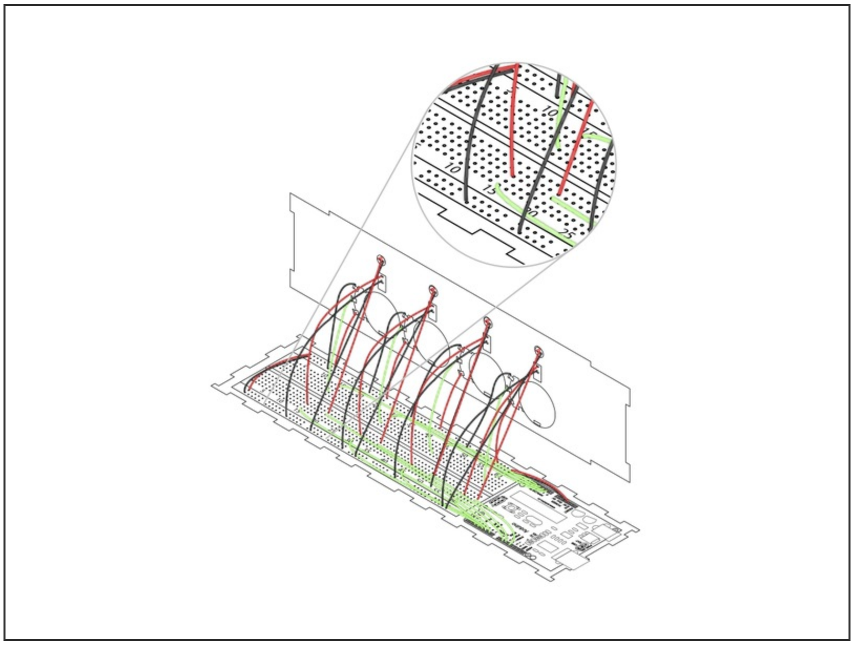
- Fit the box lid into the slots on the rear top of the box sides. The box sides are not shown in the illustration to provide a clearer view of the connections.
- Connect the knobs first as they are at the back of the box. Connect each knob’s red lead to the red power rail on the protoboard and the black leads to the black power rail. These can be positioned anywhere on the rail, but placing them in the closest row will help keep the wires tidier.
- From left to right connect the green lead on the first knob to row 15 and the subsequent knobs' green leads to the other rows on the protoboard which are connected to the ANALOG pins on the Arduino.

Step 10 — Hook up the switches



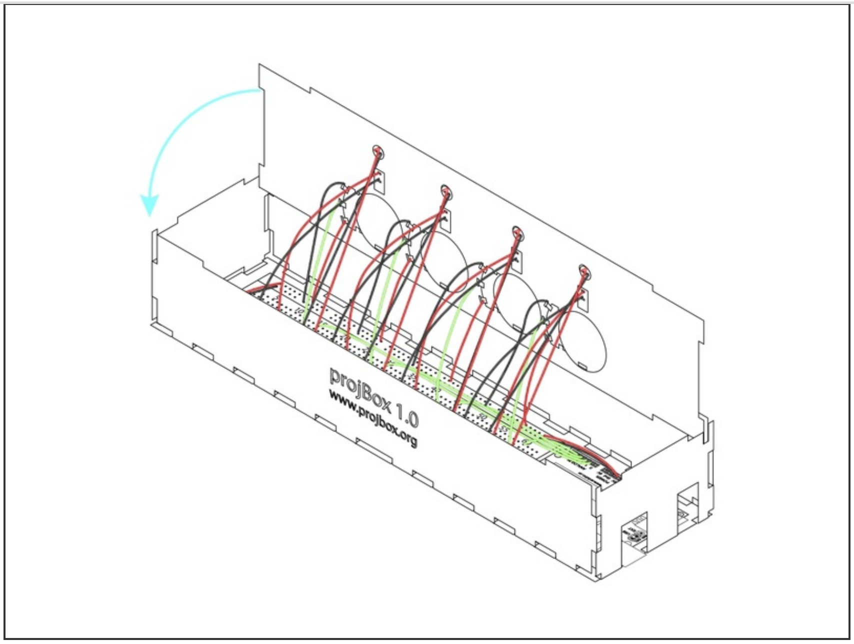
- Connect the black lead of each switch to the blue rail on the bottom half of the protoboard. Connect the red leads to their respective rows on the protoboard starting with row 15. These leads connect to the rows with jumpers connecting to DIGITAL 2 through 5 on the Arduino.

Step 11 — Hook up the LEDs



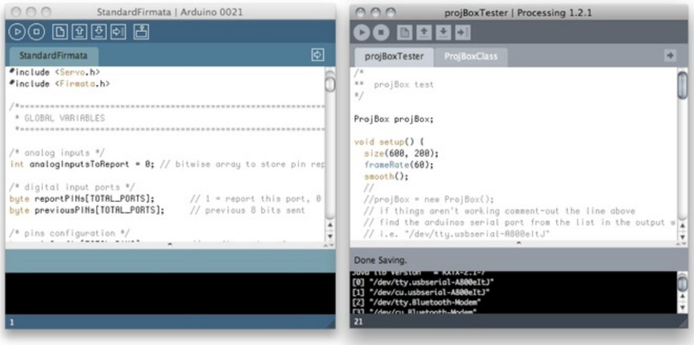
- At this point there will be a lot of wires connected inside the box. Carefully move the connected wires out of the way when adding other wires, thinking in terms of how to fold the wires in to close the lid. The main thing to look out for is the connections on the protoboard from the components on the lid. You might want to periodically go around to each wire while assembling everything and use the needle-nose pliers to make sure each connection is snug.
- To connect the LEDs run the black wires to the blue rail on the bottom half of the protoboard. Connect the red leads on the LEDs to their respective rows starting with row 20. These are the rows with jumpers connecting to DIGITAL 6, 9, 10 and 11 on the Arduino.

Step 12 — Lower the lid



- Closing the lid is the final step in finishing the projBox controller. This is possibly the trickiest part of the process as the wires have a natural springiness and will try to push the lid back open unless they are wrangled just right.
- Start by tucking back each lead wire running from the box lid. There should be enough play to be able to bend them a bit somewhere toward the middle. Lift the lid gently out of the back slots and then start pushing the lid down. If you bent the wires right they should fold in and the lid should be able to lower entirely in place. This may take a bit of trial and error. Again, using the pliers to ensure that the connections are firm is really useful.
- Be patient with this part of the process. Try and try again until everything fits together nicely.
- Once the lid fits down, wrap a bit of electrical tape around the box to hold the lid in place. In the next step we'll plug the projBox into a computer via USB and run some test software to make sure everything is working. After that you can find a way to more permanently attach the box lid, like using more tape.

Step 13 — Install the Arduino and Processing IDEs


StandardFirmata | Arduino 0021
StandardFirmata
1
projBoxTester | Processing 1.2.1
projBoxTester
ProjBoxClass
21

```
#include <Servo.h>
#include <Firmata.h>

/*==========================================================
 * GLOBAL VARIABLES
 *==========================================================

/* analog inputs */
int analogInputsToReport = 0; // bitwise array to store pin rep

/* digital input ports */
byte reportPINs[TOTAL_PORTS];       // 1 = report this port, 0
byte previousPINs[TOTAL_PORTS];     // previous 8 bits sent

/* pins configuration */
```

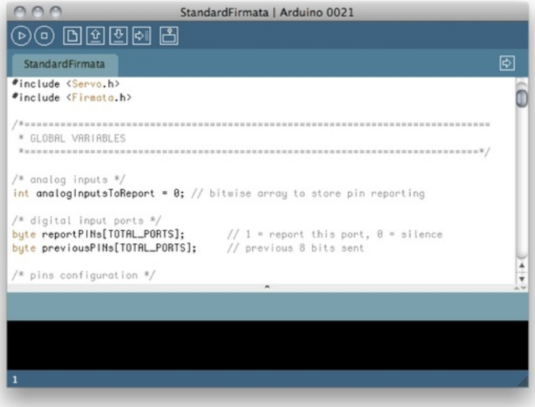
```
/*
**  projBox test
*/

ProjBox projBox;

void setup() {
  size(600, 200);
  frameRate(60);
  smooth();
  //
  //projBox = new ProjBox();
  // if things aren't working comment-out the line above
  // find the arduinos serial port from the list in the output w
  // i.e. "/dev/tty.usbserial-A800eltJ"
```

Done Saving.

```
Java lib Version   = RXTX-2.1-7
[0] "/dev/tty.usbserial-A800eItJ"
[1] "/dev/cu.usbserial-A800eItJ"
[2] "/dev/tty.Bluetooth-Modem"
[3] "/dev/cu.Bluetooth-Modem"
```

- You will need both the Arduino and Processing development environments installed on your computer to get the test software going.
- Instructions for installing the Arduino IDE (Integrated Development Environment) and communicating with the Arduino board can be found here.
- A guide for installing the Processing IDE is available here.
- Make Magazine has also published excellent guides for both technologies written by the founders of these now-global initiatives: Getting Started with Processing is by Ben Fry and Casey Reas, and Getting Started with Arduino is by Massimo Banzi.

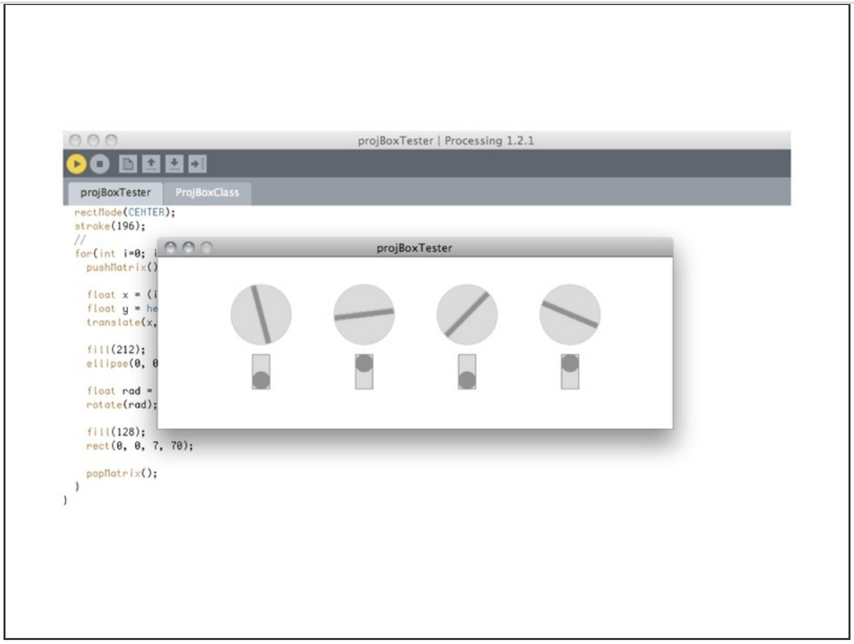
Step 14 — Upload Standard Firmata to the Arduino



```
#include <Servo.h>
#include <Firmata.h>

/*==============================================================================
 * GLOBAL VARIABLES
 *============================================================================*/

/* analog inputs */
int analogInputsToReport = 0; // bitwise array to store pin reporting

/* digital input ports */
byte reportPINs[TOTAL_PORTS];       // 1 = report this port, 0 = silence
byte previousPINs[TOTAL_PORTS];     // previous 8 bits sent

/* pins configuration */
```

- Once you've got both programs up-and-running download the projBoxTester software sketch from http://projbox.org/software. It contains the Processing source code for an application that displays a graphical representation of the projBox that is connected.
- For the application to work you must first load Standard Firmata onto the Arduino board inside the projBox. First connect a USB cable from the projBox USB port to a USB port on your computer.
- Then, in the Arduino IDE go to **File -> Examples -> Firmata** and select **Standard Firmata**. Under the **Tools -> Board** menu select the board model you are using (probably Duemillanove or the newer Uno board).
- Under **Tools -> Serial Port** select the port listed for the board. Once it is selected upload the sketch to the Arduino by clicking on the icon at the top left of the Arduino sketch window (the one with the arrow pointing right). A message will appear at the bottom left of the window saying "Upload Complete" when it is done. Once Firmata has been uploaded you can close the Arduino IDE entirely, as the remaining steps will be done using only the Processing IDE.

Step 15 — Run the projBoxTester software

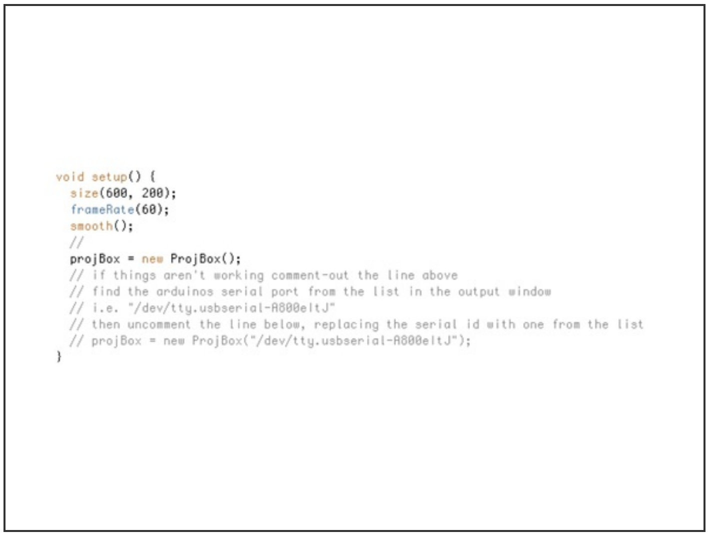


```
  rectMode(CENTER);
  stroke(196);
  //
  for(int i=0; i
    pushMatrix()

    float x = (i
    float y = he
    translate(x,

    fill(212);
    ellipse(0, 0

    float rad =
    rotate(rad);

    fill(128);
    rect(0, 0, 7, 70);

    popMatrix();
  }
}
```

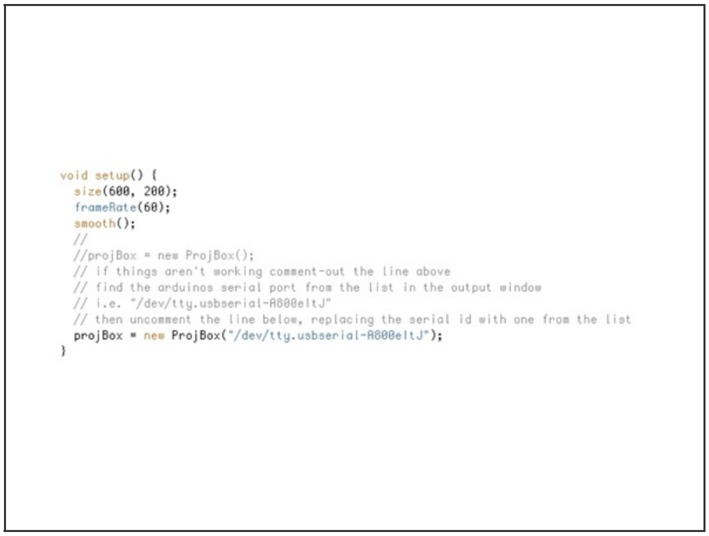
- Open the projBoxTester sketch in Processing. Press the **Play** button and the software will compile and run instantly.
- If the application appears as in the image shown here, then great! If it doesn’t, take a look at the troubleshooting tips in the next step to get the Arduino and Processing talking.
- With everything working on the software side, try turning the knobs and flipping the switches on the projBox itself. If the wires inside the box are connected properly the knob and switch graphics on the screen will change along with the physical ones. By default the LEDs on the box will mimic the on/off state of the switches.
- It might be that something doesn’t behave as you expected, like a switch not working or a knob turning the wrong direction on the screen. If so, open the lid of the projBox and get back to re-connecting the wires utilizing the steps listed above until everything works.
- That’s it! Once everything is working as you expect it to you may want to attach the lid a bit more solidly. I usually just wrap some electrical tape around the box in a few places.

- From here you'll be ready to write custom software in Processing that you can control using the projBox. A bit more detail about the ProjBox software class and how to use it in your own Processing code can be found at http://projBox.org/software.

Step 16 — Troubleshooting the projBoxTester



```
void setup() {
  size(600, 200);
  frameRate(60);
  smooth();
  //
  projBox = new ProjBox();
  // if things aren't working comment-out the line above
  // find the arduinos serial port from the list in the output window
  // i.e. "/dev/tty.usbserial-A800eltJ"
  // then uncomment the line below, replacing the serial id with one from the list
  // projBox = new ProjBox("/dev/tty.usbserial-A800eltJ");
}
```



```
void setup() {
  size(600, 200);
  frameRate(60);
  smooth();
  //
  //projBox = new ProjBox();
  // if things aren't working comment-out the line above
  // find the arduinos serial port from the list in the output window
  // i.e. "/dev/tty.usbserial-A800eltJ"
  // then uncomment the line below, replacing the serial id with one from the list
  projBox = new ProjBox("/dev/tty.usbserial-A800eltJ");
}
```

- Often when the software is first run there will be an error. This is most likely because the software is trying to connect to another piece of hardware instead of the Arduino.
- To fix this, change the `setup()` function of the code. The original looks like figure 1 in this step.
- When you first run the program it will show a list of serial devices in the Processing IDE's output window. Find the serial device that connects to your Arduino board. It will be named the same thing that it was in the Arduino IDE when you selected it under **Tools -> Serial Ports**. Copy the name of the device and paste it into the code, uncommenting and commenting out the necessary lines, as seen in figure 2 of this step.
- After passing the serial device name to the `ProjBox()` constructor try running the projBoxTester again by clicking the **Play** button. It should work. If it still doesn't work, check your USB connection. Beyond that you may want to look at the Arduino documentation, ask questions on a forum dedicated to the topic, visit a local hack lab or seek the advice of a knowledgable friend.

This document was last generated on 2012-11-02 01:46:19 AM.